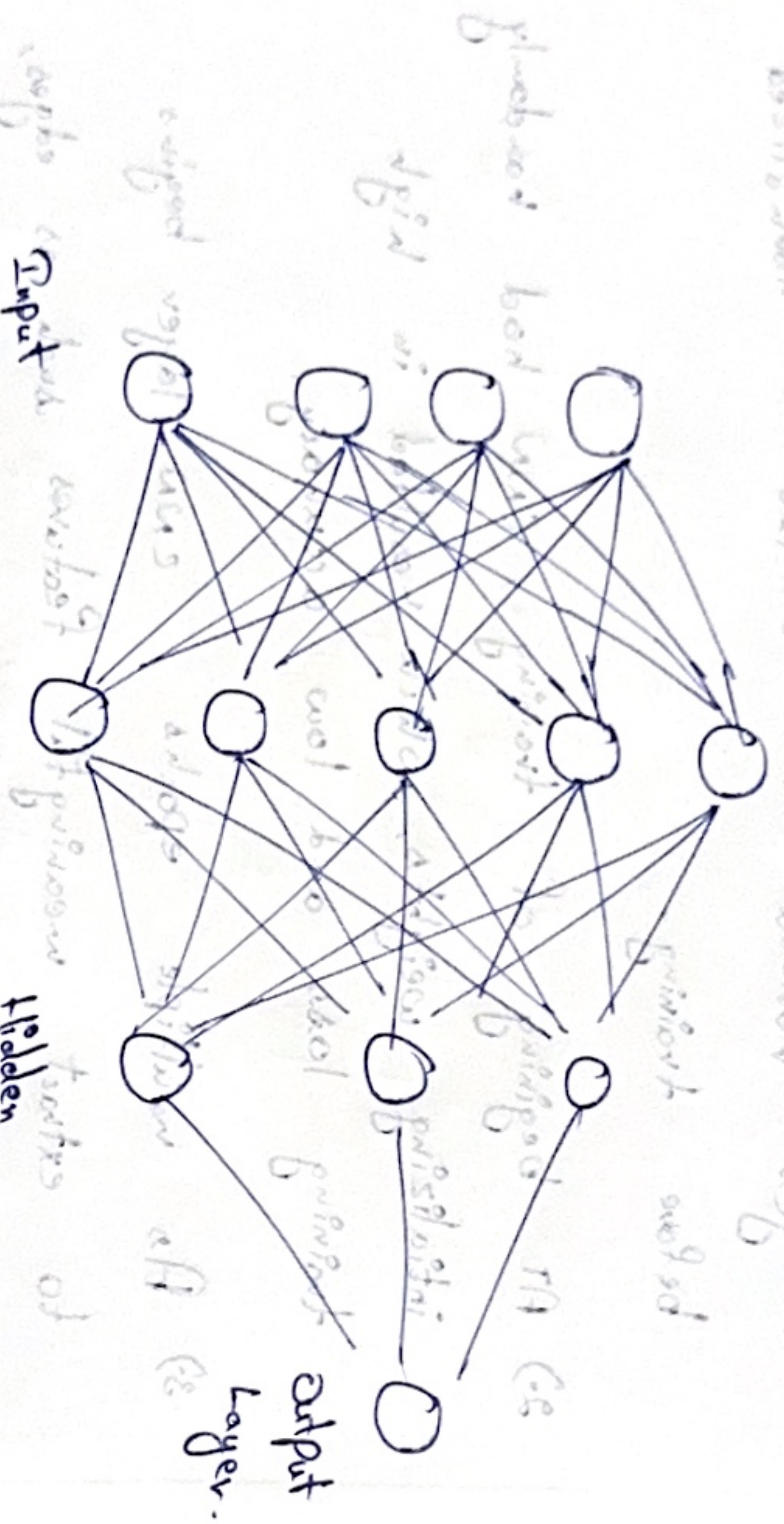


RNN Architecture

• objectives

• apply two steps to given input

• loss function to get a set of better output



Input layer

Hidden layer

Output layer

philosophy with other best neural network

and also, also, loss function

• learning process

• is this

• ideas of how we build - sequentially

• growth of the

Exp 1:-

Build a Recurrent Neural Network.

Aim:-

To build train a recurrent neural network (RNN) for sequence modelling.

Objectives:-

- To understand the working principles of RNN
- To preprocess sequential data for RNN
- To design and implement an RNN using Python.
- To train the RNN model and evaluate the performance.
- To analyze the output.

Pseudocode:-

- 1) Start.
- 2) Import necessary libraries.
- 3) Load dataset.
- 4) Preprocess dataset.

→ clear data

→ Taking / encode sequences

→ Pad / truncate sequences

→ Split into training and validation sets.

5) Define RNN model.

→ Initialize sequential model.

→ Add Embedding layer or input layer.

→ Add simple RNN layers (3).

Gradient Descent + Gradient Descent

1) Gradient Descent + Gradient Descent

Gradient Descent

Gradient Descent

2) Gradient Descent + Gradient Descent

Gradient Descent

3) Gradient Descent + Gradient Descent

Gradient Descent

4) Gradient Descent + Gradient Descent

Gradient Descent

Gradient Descent

5) Gradient Descent + Gradient Descent

Gradient Descent

Gradient Descent

Gradient Descent

Gradient Descent

Gradient Descent

6) Gradient Descent + Gradient Descent

Gradient Descent

Gradient Descent

7) Gradient Descent + Gradient Descent

Gradient Descent

6) Compile model.

- Specify optimizer.
- Specify loss function.
- Specify evaluation.

7) Train model.

- Feed training data.
- Specify batch size and no. of epochs.
- Validate any validation set.

8) Evaluate model.

- Test on unseen data.
- Print accuracy / loss.

9) End.

Output:-

Accuracy = 0.7174.

Macro Precision = 0.7197.

Macro Recall = 0.7174.

Macro F1 Score = 0.7167.

Precision = [0.74181495 0.6975290]

Recall = [0.66704 0.76784]

F1 = [0.70244313 0.730992]

John Wiley (4)

positively related

positively related

positively related

John Wiley (7)

positively related

positively related

positively related

John Wiley (8)

positively related

positively related

John Wiley (9)

John Wiley

positively related

positively related

positively related

positively related

positively related

positively related

positively related

Confusion Matrix :-

[8338 4162]

[2902 9595]

classification report :-

neg Precision recall Score Support

Pos 0.74 0.67 0.70 12500

0.77 0.73 12500

Accuracy

weighted avg 0.72 0.68 0.72 13500

weighted avg 0.72 0.67 0.73 13500

John Wiley